Towards an Approach for Improving Exploratory Testing Tour Assignment based on Testers' Profile

Letícia De Souza Santos[1], Rejane Maria Da Costa Figueiredo[1,2][a], Rafael Fazzolino Pinto Barbosa[1], Auri Marcelo Rizzo Vincenzi[3], Glauco Vitor Pedrosa[1,2][b] and John Lenon Cardoso Gardenghi[1,2][c]

[1]*Information Technology Research and Application Center (ITRAC), University of Brasilia (UnB), Brasilia, DF, Brazil*

[2]*Post-Graduate Program in Applied Computing, University of Brasilia (UnB), Brasilia, DF, Brazil*

[3]*Federal University of Sao Carlos (UFSCar), Sao Carlos, SP, Brazil*

Keywords: Recommendation Systems, Exploratory Testing, Profile of Testers.

Abstract: This work presents an empirical study on the relationship between the testers' profile and their efficiency and preference in the application of tours with tourist metaphor for exploratory software testing. For this purpose, we developed and applied a questionnaire based model to gather as much as possible information about the knowledge, expertise and education level from a group of testers. The results indicated that, in fact, the testers' profile have impact on the application of tours used in the tourist metaphor: there are differences between the tours preferred by different levels of education and most of testers tend to choose those tours based on what they believed to have the shortest execution time. This work raises a valuable discussion about a humanized process of assigning test tasks in order to improve the efficiency of software testing.

1 INTRODUCTION

Software testing is an arduous and expensive activity. In this sense, there are opportunities for developing strategies to reduce the test execution time and to increase defect findings. One approach is to allocate test cases according to the testers profile in a way to maximize testing productivity. However, optimizing the allocation of manual test cases is not a trivial task: in large companies, test managers are responsible for allocating hundreds of test cases among several testers.

Studies such as (Anvik et al., 2006) and (Miranda et al., 2012) have shown that it is possible to employ recommendation systems to allocate tasks to specific profiles based on the analysis of previous allocations. So, a recommendation system can assist in assigning test to a team of testers, seeking to contribute to the teams' productivity.

Through the Exploratory Testing (ET), it is possible that the tester does not depend on a set of predesigned test cases, as ET contains the steps of design and test execution. To systematize the ET process, the work in (Whittaker, 2009) proposed the Tourist Metaphor, which draws an analogy between software testing and a tourist visiting in a city. Tourism is a mixture of structure and freedom, just like the Exploratory Test.

The tester's profile influences the application of tours used in the Tourist Metaphor (Miranda et al., 2012). In this sense, a way to maximize productivity in a test team is to allocate tasks according to the testers profile. This is due to the understanding that the generated test cases and their sequences vary a lot depending on the tester.

This paper presents an empirical study to support the identification of testers' profile for the use of the Exploratory Testing approach considering the Tourist Metaphor. A group of 60 testers was interviewed to gather information about their education level, expertise, computational knowledge and preference among those tours considered in the tourist metaphor. The idea is to raise correlations between this information in order to develop a test recommendation system.

In summary, the main contributions of this work are:

- Literature review: we raised references on the relationship between the testers' profile and the assignment of test tasks;

[a] https://orcid.org/0000-0001-8243-7924
[b] https://orcid.org/0000-0001-5573-6830
[c] https://orcid.org/0000-0003-4443-8090

Santos, L., Figueiredo, R., Barbosa, R., Vincenzi, A., Pedrosa, G. and Gardenghi, J.
Towards an Approach for Improving Exploratory Testing Tour Assignment based on Testers' Profile.
DOI: 10.5220/0011113800003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 2*, pages 183-190
ISBN: 978-989-758-569-2; ISSN: 2184-4992

- The development of a questionnaire for tester's profile identification;
- Experimental study on the relationship between the testers' and their efficiency in the application of tours in the context of Exploratory Tests with the Tourist Metaphor;
- Data analysis to support the definition of a Recommendation System for automatic assignment of test tasks with the Tourist Metaphor.

The remaining of this paper is organized as follows: Section 2 discusses related works and presents some definitions concerned to our work; Section 3 describes the methodology adopted in our work; Section 4 presents the profile of the group of professionals involved with our data collection; Section 5 presents the results and discussions and Section 6 finalizes with the conclusion and future works.

2 BACKGROUND

The theoretical background of this study is based on the software testing and the exploratory test with the tourist metaphor. In the following, we present some definitions and correlated works that motivated the development on this work.

2.1 Software Testing

The testing activity is essential for Software Engineering, as it is a tool to ensure the quality of the software product from the identification of failures during its development (Myers et al., 2004) when fault correction is cheaper. The *feedback* of real behaviors makes testing a fundamental quality assurance analysis technique in the industry, although it may require a lot of human work and the scientific community considers more the use of automated tests (Bertolino, 2007).

Testing by human testers is relevant to real-world software development as it allows the identification of new BUGs, especially in interactive systems. Human testers have advantages over machines, given their capacity for knowledge, learning and adaptation to new situations, which facilitates the process of efficient recognition of problems (Itkonen et al., 2015).

Through the Exploratory Testing (ET), it is possible that the tester does not depend on a set of predesigned test cases, since this test approach contains the steps of design and test execution, in which testers are constantly learning and adapting activities. In a practical way, the tester learns iteratively about the product and its failures, designs and executes the tests dynamically and systematically (Whittaker, 2009).

2.2 Exploratory Test with the Tourist Metaphor

To systematize the ET process, the work in (Whittaker, 2009) presented the Tourist Metaphor, which draws an analogy between software testing and a tourist visiting in a city. According to (Whittaker, 2009), tourism is a mixture of structure and freedom, just like the exploratory test.

In the analogy presented, the software features are separated into "districts", that one decides to explore. Each district has a set of "tours", which represent the different ways of going trough the characteristics and functionalities of the software. This analogy helps the test team to communicate about what should be tested and how.

In this work, we considered 23 tours present in 6 districts. Among these 23, 16 were used in the interview with testers. Therefore, the descriptions of the 16 tours chosen by the participants are presented in the following.

- Tours Trough the Business District
  - *Intellectual Tour*: Run the software with inputs so that it operates under conditions of maximum load or that require more processing.
  - *Landmark Tour*: Determines what are the main characteristics of the product (reference points) and in which order to visit them.
  - *Garbage Collector's Tour*: Test all menu items, all error messages, among others, and to visit each one in a methodical way, going through the shortest path.
  - *Guidebook Tour*: Follow the user manual.
  - *FedEx Tour*: Look for identifies where the data is changed in order to assess if they are being corrupted on the way.
- Tours Trough the Historical District
  - *Bad-neighborhood Tour*: Visit areas of code full of defects. Focus test effort on areas with the highest concentration of defects.
- Tours Trough the Entertainment District
  - *Back Alley Tour*: Visit the less attractive functionalities from the user's point of view.
  - *Supporting Actor Tour*: Regardless of the salespeople's efforts, the customer often ends up being more interested in peripheral characteristics than the main ones.
  - *All-nighter Tour*: Challenge the software seeking to popularize the same data and force consecutive readings and writes of the values of the variables.

- Tours Trough the Tourist District
  - *Supermodel Tour*: Look for superficial defects in the software product related to its appearance.
  - *Collector's Tour*: Visit every possible location of the software and document every output obtained.
- Tours Trough the Hotel District
  - *Couch Potato Tour*: Work as little as possible. Even if the tester isn't doing much, it doesn't necessarily mean that the software isn't.
  - *Rained-out Tour*: Identify a list of time-consuming operations to perform. Start an operation and then stop it.
- Tours Trough the Seedy District
  - *Antisocial Tour*: Do the opposite of what is expected in the software.
  - *Saboteur*: Limit access or exclude required resources.
  - *Obsessive-Compulsive Tour*: Repeat, redo, copy, paste, borrow, the same action several times in a row.

2.3 Impact of Testers' Profile in Software Testing

As a human-based activity, the results of a software product test are dependent on human factors and pose challenges for software development teams, such as the search for a more effective way to increase testers' motivation and satisfaction (Deak et al., 2016).

For the past 50 years, Software Engineering has been concerned with the influence of human personality on individual work tasks, as the systematic literature review done by (Cruz et al., 2011) points out. In fact, some works on exploratory tests conclude that the human personality can influence this test method (Bach, 2003; Whittaker, 2009; Itkonen et al., 2015; Itkonen et al., 2012; Shoaib et al., 2009). The actions of testers during the application of exploratory tests can vary significantly from one person to another, that is, the methodology adopted is directly related to the personality traits of each tester (Shoaib et al., 2009).

The experiment carried out by (Shoaib et al., 2009) was designed to identify the testers who can achieve the best result during the application of exploratory tests. The results showed a positive relationship between the exploratory test and human personality traits. In addition, organizations have adopted alternative methodologies and workforces to efficiently deliver software (Dubey et al., 2017).

The work in (Berner et al., 2005) claims that automated testing can never completely replace manual testing. (Martin et al., 2007) presents reports which state that the problems related to software testing in the industry involve the company's socio-technical environment and organizational structure.

The relationship between software testing and the human aspect was studied by (Shah and Harrold, 2010) in the context of a service-based company. The results showed that the attitudes of older professionals can significantly influence the attitudes of more inexperienced people.

In this sense, as already addressed by (Cruz et al., 2011), the team's performance may vary depending on its members and their personalities and experiences. This contributes strongly to the present work, since the laboratory involved is an environment composed of team members at different levels of education and experience and in continuous *turnover*.

Based on the literature reviewed and the techniques adopted in previous studies, this work seeks to apply concepts related to the profile of the tester in order to gather data to develop a recommendation system for case test based on the Tourist Metaphor.

3 METHODOLOGY

The methodology adopted by this work comprises four basic phases: (1) research planning, (2) data collection, (3) data analysis, and (4) reporting the results.

The interviews were developed with undergraduate course of Software Engineering at our university, engaging 40 participants, and a specialization course in the area of Computer Science, involving 20 participants. These participants were the object of study. In the data collection phase, the research procedures employed were: documentary research; bibliographic research; and action research. A questionnaire was applied to each participant, in order to identify their characteristics.

The process of digital transformation in the context of this work suggests a large number of low complexity services, which allows for a large number of test cycles in a short period of time. This characteristic favors learning and evolution related to testing activities.

Figure 1 presents the continuous learning process, together with the methodology proposed for this work. The left part of Figure 1 refers to the Analysis and is linked to the collection of the profile of the tester and the analysis of the data collected after each test cycle. The right part of Figure 1 lists what is intended to be used as an experiment, which makes it

possible to record the test cycles and propose Attribution.

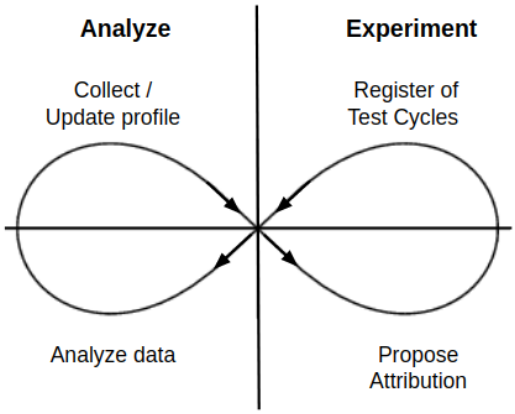

Figure 1: Approach to the Recommendation System.

The profile of each tester will be drawn from the information extracted from the questionnaire applied.

The recommendation strategy suggested in this research can use information from the preferences for each *tour* by the participants of the test dynamics and, thus, determine which *tours* are most suitable for certain testers, based on the history of tests carried out and reported preferences.

It is important to highlight that the execution of a *tour* by a tester gives rise to a set of tests generated from the *tour*. The assignment of *tours* and testers should be done dynamically, with a learning process during each test cycle. In other words, the attribution of *tours* and testers must consider, in addition to characteristics, the learning obtained during the previous test cycles. Learning can maximize the efficiency of applying test cases. In this context, the word ”efficiency” refers to the failure identification rate during the application of the test cases it is given by Equation 1.

$$\text{efficiency} = \frac{\text{number of failures identified}}{\text{number of test cases of the tour}} \quad (1)$$

In this work, as proposed in (Miranda et al., 2012), the assignment is represented throughout the text as a set of test and tester pairs, represented by $\{CTn;Tn\}$ (Test Case n and Tester n). The user is represented by a typical test manager and recommendations are made by comparing a specific test case with the profile of a tester.

A correlation is found between the efficiency in the test process and the different variables that make up the profile of the tester. This profile will be identified based on different questions answered by the testers in a digital questionnaire.

After the questionnaires are answered, the next step is to collect data on the efficiency of each tester in carrying out the test activities. To collect this data, correlation tests are performed to identify whether there is any relationship between the profile variables and the efficiency in the tests of a given tester or the testing team as a whole.

The data explain the impact of the variables of the profile of the tester on the efficiency of the tests. After this stage, it is possible to use the questionnaire to make a correlation test between the variables surveyed.

4 DATA COLLECTION

The interviews to collect information on the testers' profile were conducted with professionals and students engaged with our university. In summary, the collected data contains information from:

- 40 undergraduate students of Software Engineering
- 20 professionals of a specialization course in the area of Computer Science

The collected data interviews were developed with undergraduate course of Software Engineering at our university, engaging 40 participants, and a specialization course in the area of Computer Science, involving 20 participants. These participants were the object of study. In the data collection phase, the research procedures employed were: documentary research; bibliographic research; and action research. A questionnaire was applied to each participant, in order to identify their characteristics.

To extract information to support an automatic test task assignment process, the action research procedure was selected. Therefore, activities were carried out in a participatory and interactive way by the researcher and the participants.

The following steps were taken: identification and survey of personal characteristics to be addressed in the questionnaire; extraction of information about the personal characteristics of each tester; monitoring the test dynamics of a fictitious service created with different characteristics; data analysis; and proposal of a filtering strategy to consider future recommendations.

It is possible that the participants with different background knowledge and experiences in tests are not considered when assigning test tasks performed using Exploratory Tests. Therefore, students who are at the beginning of their undergraduate course and have no experience in tests, can be allocated to test tasks that have the same level of difficulty as test tasks allocated to experienced testers.

The questionnaire applied in this study was based on the *survey* conducted by (Geras et al., 2004). Based on this, in order to characterize software tests and quality assurance practices, the questionnaire was divided into two categories: (i) personal and professional profile; and (ii) specific knowledge about testing;

The first category was developed to collect personal information and the professional profile of the respondent, with an interest in understanding their level of education, undergraduate courses and experiences with programming languages.

The second category sought to evaluate the degree of familiarity of the testers with the test activity. The questionnaire addressed techniques and testing criteria, for example, to assess the experience of testers in this regard. In an application of the recommendation process proposed in this research, a third category could encompass an open question, which asks the tester to report their experience. The research team intends to include this category in future works.

A fictitious service was developed so that testers could make *tours* from the Tourist Metaphor and find as many defects as possible. In order for the participants to perform the test, a *Guide for the Application of Exploratory Tests* was created. All the participants already had minimal knowledge about the Exploratory Testing approach.

5 RESULTS AND DISCUSSIONS

The information collected from the questionnaire is presented in graphic form, and refers to level of education; technologies most used; experience in Software Testing; experience in Application Testing; experience in Exploratory Tests; and experience in each test phase.

Figure 2 reveals the distribution of participants involved in this research in relation to the level of education. These levels varied considerably within the three participating groups, and the largest number compromised postgraduate students.

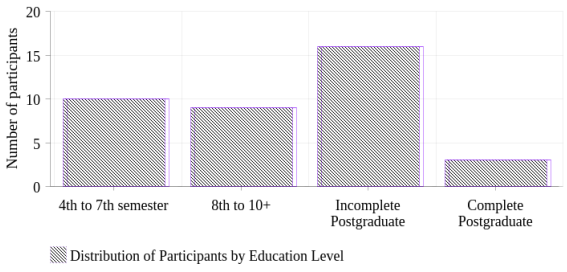

Figure 2: Distribution of Participants by Education Level.

The performance of tests considering different professional expertise was an important information gathered in our study, as some of the participants have already had experiences in the labor market, while others are in the middle of the Software Engineering course, which decreases their level of expertise in comparison to graduate participants.

Figure 3 presents the techniques that the participants had some knowledge of or were skillful in. It help us to predict how testers would be able to perform tests that would require further exploration of software or, in the case of this research, a government service.

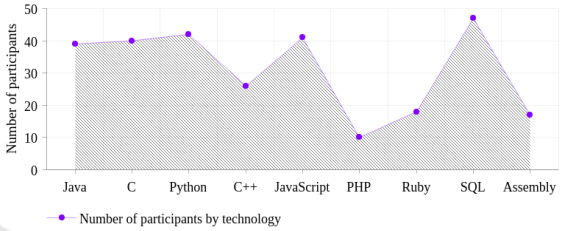

Figure 3: Number of Participants by Technology.

Figure 4 presents the test phases in which the three participant groups have greater experience. In the software testing discipline, the students' greatest experience was concentrated in Unit Tests, which had already been practiced by 90% of the respondents. Approximately 57% of the respondents declared to have experience in Acceptance Tests, and 33%, in Integration Tests. Meanwhile, about 20% declared experience in System Testing and only 10% declared to have no experience in any test phase.

All respondents in the Experimental Software Engineering discipline declared that they had already carried out unit tests, while 60% had already taken Acceptance tests, 40% in Integration Tests and 30% in System Tests.

The diversification of the level of education allows us to understand the different ways of looking at the software and its possible defects. This statement becomes more evident in the discussion about the answers obtained after performing the tests in the fictitious service.

The graduate class showed a level of 70% in unit test knowledge, 50% in both integration tests and system tests and 20% in acceptance tests. Also, 15% said they had not worked in any of the phases indicated.

The results of the dynamics also showed the testing techniques in that the members of each discipline have experience. In the Software Testing discipline, about 87% of the participants had already performed tests using the Functional Test technique (black box test), 57% had already used the structural test tech-

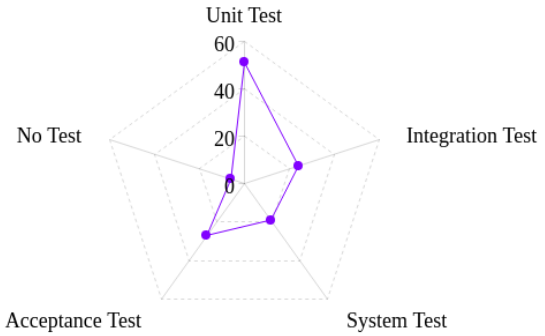

Figure 4: Level of experience in testing phases of the members of each discipline.

nique (white box), 23% had performed a defect-based testing technique, and 3% had never used any of the testing techniques presented.

Regarding the Experimental Software Engineering class, all the respondents declared that they already had experience in some of the testing techniques. 90% declared to have used the functional test technique, 30% declared to have used structural test and 10% had already used defect-based testing.

In general, respondents in the graduate course had more experience in functional testing (70%). With 50% positive responses, the structural test was the second technique mastered by students of the discipline, while 30% had already tested it based on defects. 15% stated that they had never performed the testing techniques.

Table 1 shows the results of the *tours* most used by students involved in the dynamics of Tests. The most used one was the anti-social *tour*, done by 50 of the 60 students who participated in the dynamic. This tour had already been noticed in the work of (Blinded Author(s), 0000), which presents a number of tests and failures identified by *tour*, and reports the creation of a process for validating services produced by digital transformation.

A ranking was made with the *tours* most used in each of the disciplines. Group 1 participants showed greater interest in Antisocial, Couch Potato and Obsessive-Compulsive *tours*.

It is possible to relate the choices of *tours* with the profiles outlined by the questionnaire, given the levels of knowledge. Group 2 and 3 participants presented more consistent answers about their knowledge of the types, phases, techniques, criteria and testing tools. Their answers to the question were more complete than those of Group 1.

The maturity with which the dynamics were treated by the groups also influenced the data obtained when taking the educational level into account. Group 1 presented a large number of responses in the dynamics; however, not all participants answered the profile questionnaire. In addition, many of the defects found were more related to an ad-hoc way that students ended up testing, than to a way of testing defined by some *tour*, for example.

Table 1: Table with List of Tours Used by Testers.

Selected Tours	Group 1	Group 2	Group 3	Total
Antissocial	30	6	12	48
Couch Potato	30	4	8	42
Obsessive-Compulsive	23	3	4	30
Intelectual	18	0	10	28
Supermodel	17	4	4	25
Collector	18	2	3	23
Landmark	16	4	2	22
Saboteur	11	5	5	21
Garbage Collector	14	1	3	18
Guidebook	15	0	2	17
Back Alley	7	1	1	9
Rained-out	5	0	3	8
FedEx	8	0	0	8
Supporting Actor	3	1	0	4
Bad-neighborhood	3	0	0	3
All-nighter	0	1	1	2

With the application of the Test dynamics in the three participating groups, it was possible to perceive the preferences in relation to the *tours* of the Tourist Metaphor. This information was necessary to support the creation of a test task recommendation system that is based on the Exploratory Testing approach.

All the 30 Software Engineering undergraduate participants in the Software Testing class (Group 1) made use of the *Antissocial Tour* and the *Couch Potato Tour*, and 77% chose the *Obsessive-Compulsive Tour*. The others demonstrated a more widespread interest among the listed tours, as shown in Table 1 .

Of the 10 Software Engineering undergraduate respondents in the Experimental Software Engineering discipline (Group 2), 60% opted for the *Antisossial Tour* and 40% opted for the *Intellectual tour*, while the rest of *tours* were more spread among the students'choices.

Of the 20 participants in the graduate class (Group 3), 60% chose the *Antisocial Tour*, 50% took the *Intellectual Tour*, while the rest dissipated among the other *tours*.

It is evident that some of the *tours* had a greater adherence by the participants of the three groups. This choice is related to the description of the *tour* which, in the cases of the most listed *tours*, indicate practical ways of testing software, explain how to perform the opposite of what was expected in a functionality (Antisocial), or put more load in some field than it should be able to support (Intellectual).

The *Couch Potato Tour* was strongly preferred in

the Software Testing class, which consequently has the majority of the younger and less experienced students in terms of years in Software Testing. Most of the *tours* that were little, or not chosen, required more testing time. In view of the determined time of the dynamics, the students chose the *tours* based also on what they believed to have the shortest execution time.

As shown in Figure 1, after collecting and analyzing profile data from testers, it was be possible to use the approach and, based on a future use of historical data on test cases already registered, propose an assignment of test tasks using the *tours* that were *ranked* by the subjects.

Although there are differences between the *tours* preferred by different levels of education, the ranking presents options that could be proposed to different levels of knowledge about tests, as shown by (Blinded Author(s), 0000), who carried out his experiment with a team composed mostly of undergraduate students.

After assigning the best *tours ranked*, it is possible to execute and record the test cases in order to generate *inputs* for a future recommendation of *tours*, concluding the first cycle of a continuous process proposed in this work and presented in Figure 1.

6 CONCLUSIONS AND FUTURE WORKS

This study aimed to identify profiles of testers to support the creation of a test task recommendation system based on the Exploratory Testing approach with the Tourist Metaphor. For this, we sought to gather as much relevant information as possible to assign test tasks based on the profile of the testers.

The information comes from both literature review and empirical analysis, with a sample from three groups of different levels of education, related to IT and linked to the academy. This enabled the collection of information about profiles and the achievement of testing dynamics based on the Exploratory Testing approach.

The personal characteristics of each tester influence his work with software tests and define a basic strategy for structuring a test process that is based on human characteristics in order to direct the attribution of test tasks. This strategy should consider both the test history of each tester and their profile, which are incremented with each test cycle.

This study raises a valuable discussion about a humanized process of assigning test tasks in order to generate data for the definition of a recommendation system for automatic assignment of test tasks based on the profile of each tester. In addition to testing tasks, this strategy can be extended to development contexts, given that the profile of each developer, and tester, can also influence the effectiveness of the activity and the degree of satisfaction of the developer.

It is possible to highlight two main future works derived from this research. The first one is to extract the profiles with more testers as a sample, in order to follow the exploratory testing process carried out, to build a consistent database on profiles. From a more solid database, the second future work is to apply artificial intelligence algorithms for automatic assignment of test tasks based on the profile of testers.

Finally, consolidate the implementation of a recommendation system for assigning test tasks based on the testers' profile.

REFERENCES

Anvik, J., Hiew, L., and Murphy, G. C. (2006). Who should fix this bug? In *Proceedings of the 28th international conference on Software engineering*, pages 361–370. ACM.

Bach, J. (2003). Exploratory testing explained.

Berner, S., Weber, R., and Keller, R. K. (2005). Observations and lessons learned from automated testing. In *Proceedings of the 27th international conference on Software engineering*, pages 571–579. ACM.

Bertolino, A. (2007). Software testing research: Achievements, challenges, dreams. In *2007 Future of Software Engineering*, pages 85–103. IEEE Computer Society.

Blinded Author(s) (0000). Blinded title. In *Blinded Conference*, pages 00–00.

Cruz, S. S., da Silva, F. Q., Monteiro, C. V., Santos, P., and Rossilei, I. (2011). Personality in software engineering: Preliminary findings from a systematic literature review. In *15th annual conference on Evaluation & assessment in software engineering (EASE 2011)*, pages 1–10. IET.

Deak, A., Stålhane, T., and Sindre, G. (2016). Challenges and strategies for motivating software testing personnel. *Information and software Technology*, 73:1–15.

Dubey, A., Singi, K., and Kaulgud, V. (2017). Personas and redundancies in crowdsourced testing. In *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*, pages 76–80. IEEE.

Geras, A. M., Smith, M. R., and Miller, J. (2004). A survey of software testing practices in alberta. *Canadian Journal of Electrical and Computer Engineering*, 29(3):183–191.

Itkonen, J., Mäntylä, M. V., and Lassenius, C. (2012). The role of the tester's knowledge in exploratory software testing. *IEEE Transactions on Software Engineering*, 39(5):707–724.

Itkonen, J., Mäntylä, M. V., and Lassenius, C. (2015). Test better by exploring: Harnessing human skills and knowledge. *IEEE Software*, 33(4):90–96.

Martin, D., Rooksby, J., Rouncefield, M., and Sommerville, I. (2007). 'good'organisational reasons for'bad'software testing: An ethnographic study of testing in a small software company. In *Proceedings of the 29th international conference on Software Engineering*, pages 602–611. IEEE Computer Society.

Miranda, B., Aranha, E. H. d. S., and Iyoda, J. M. (2012). Recommender systems for manual testing: deciding how to assign tests in a test team. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*, pages 201–210. ACM.

Myers, G. J., Badgett, T., Thomas, T. M., and Sandler, C. (2004). *The art of software testing*, volume 2. Wiley Online Library.

Shah, H. and Harrold, M. J. (2010). Studying human and social aspects of testing in a service-based software company: case study. In *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of software Engineering*, pages 102–108. ACM.

Shoaib, L., Nadeem, A., and Akbar, A. (2009). An empirical evaluation of the influence of human personality on exploratory software testing. In *2009 IEEE 13th International Multitopic Conference*, pages 1–6. IEEE.

Whittaker, J. A. (2009). *Exploratory software testing: tips, tricks, tours, and techniques to guide test design*. Pearson Education, London, England.